DNS : Moyen de défense

Fabrice Prigent

RéSIST

Mardi 28 Avril 2009

Support

Protocole support de l'Internet

- Efficace
- Déployé
- Reconnu
- Excellente mise à l'échelle

Fonction

La fonction de base est simple

- Mettre en corrélation Nom et IP

Puis s'est élargie

- Donner les MX
- Donner des services génériques
  - SRV (_ldap SRV 0 100 389 servms.site.fr)
- De plus en plus génériques
  - SPF (v=spf1 ip4 :66.135.209.192/27 all)
  - SenderID (spf2.0/pra ip4 :213.246.47.192/26 -all)

Question ?

Et si nous l'utilisions la défense ?

Préambule

C'est déjà fait

- Pour l'antispam
- Pour certains services

AntiSPAM

Les défenses en authentification

- Le SPF, et le senderID
- DKIM

Les défenses par RBL

- La facilité de mise à l'échelle (DNS spécialisé)
- rbl, rwl

Généraliste

Une société propose un service sur http://www.opendns.org

- Bloque les domaines conficker
- Bloque les sites de malware
- Mais c'est lointain, et hors de contrôle

Filtrage de domaines non productifs

- Résolution sur localhost
- Utilisation du fichier hosts de windows

Faire mieux

Mais on peut sans doute faire mieux.

Les attaques classiques utilisant le DNS

Quasiment "psychologiques" :

- Phishing
- Typosquatting
- Parasitisme

FastFlux : simple flux

Technique utilisée par les botnets pour faire résoudre le nom de domaine par les bots eux-mêmes

- Domaine réservé anonymement
- Géré par un serveur "bulletproof"
- Les IP correspondantes sont multiples et aléatoires
- Les IP sont sélectionnées par Load balancing (sur BP, entre autres)
- Souvent couplé à des proxies aveugles (vers le "mothership")

FastFlux : double flux

Evolution du FastFlux simple flux. On réplique le phénomène aux serveurs DNS

- DNS server changeant toutes les 5 minutes (ou moins)
- Ces DNS serveurs interrogent les "MotherShip" pour les réponses.
- Les serveurs sont souvent sélectionnés par BP [1]

[1] http://www.honeynet.org/node/132

HydraFlux

Evolution du FastFlux simple flux

- Le bot fait le proxy vers plusieurs "mothership".
- Le port n'est plus le port 80, mais un port aléatoire [2]

[2] http://isc.sans.org/diary.html?storyid=4753

Les buts

Buts

- Source d'infection
- Source d'ordres
- Source de mise à jour

Lutte Pré-conficker

Lutte contre les réseaux Pré-Conficker

- Repérage d'un domaine infectieux.
- Résolution sur un serveur sous contrôle.
- Renvoi d'un code "annihilant".

Conficker-A

- Blocage des DNS (encore) pour les sites antiviraux
- Requête DNS vers 250 domaines différents pour chaque jour (Rendezvous points)
- 5 TLD concernés
- Cycle de 3 heures
- Obtention du code par http ://ip.ip.ip.ip/search ?q=x&aq=7 [3]

[3] http://mtc.sri.com/Conficker/

Bataille contre Conficker-A

Plusieurs modes de défense

- Conficker Cabal : réservation des domaines
- RBL des domaines futurs, renvoi d'une IP fixe
- Blocage par les proxies des requêtes de la forme http ://ip.ip.ip.ip/search ?q=x&aq=7

Conficker-B

Même principe, mais :

- Attaque brute-force des partages
- On n'évite plus les ukrainiens. (GeoIP)
- 8 TLD concernés
- Cycle de 2 heures
- Obtention du code par http ://ip.ip.ip.ip/search ?q=x

Bataille contre Conficker-B

Même modes que Conficker-A

- Blocage par les proxies des requêtes de la forme http ://ip.ip.ip.ip/search ?q=x (Problème : google !)

Conficker-B++

- Peu de différences (16% du code)
- Le HTTP n'est plus le seul moyen de faire un update
- Les pirates sont-ils battus ?

Conficker-C

- 85% de code changé (dont MD6 !)
- Choix de 500 domaines parmi 50000
- 110 TLD concernés
- Un test par 24 heures
- Refus d'adresse IP "triviales"
- Refus d'adresses blacklistées (ironique non ?)
- Refus d'adresses IP unique pour plusieurs domaines.
- Obtention du code par http ://ip.ip.ip.ip/

Lutte contre Conficker-C

- Blocage DNS compliqué (1,5 millions de domaines pour le mois d'avril)
- Collision dans 1% de cas
- Détection par le virus
- Blocage par les proxies compliqué (pas de chaine de caractères)

Lutte contre Conficker-C

- Le but est de détecter au plus tôt les conficker.
- Regardons ce qui se passe pour le DNS.

Modification du DNS

Regardons ce qui se passe :

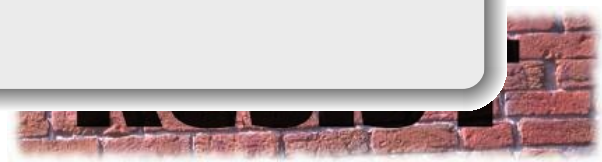
- Journalisation des requêtes (2000 / minute)
- Rotation pour éviter la saturation (1 Mo / minute)
- Suivi des requêtes

Modification du DNS

Modification du comportement du bind

named.conf

```
logging {
channel client_log {
file "/var/log/named.log" versions 10 size 100m ;
print-time yes ;
} ;
category queries {
client_log ;
} ;
} ;
```



Surveillance des journaux

Un script perl va en permanence

- Suivre le journal
- Comptabiliser le domaine demandé, et le client
- Insérer les informations dans une base de données
- Réagir à un seuil par machine

Requête de domaines jamais demandés

Après une période d'apprentissage (2 semaines), on comptabilise les domaines jamais demandés, et à partir d'un seuil (50), on remonte une alerte.
Les problèmes rencontrés :

- Les serveurs font de nombreuses requêtes.
- Apparition d'interrogations amusantes
  - nom-c73451c736d.mshome
  - dhcp-7-4-65.mshome
- Comportement associé au profil :
  - Page web avec des images provenant de 50 domaines
  - Renforcé par le phénomène de nationalité "exotique"

Requête de domaines jamais demandés : conclusion

- La détection est efficace
  - Les conficker-A et Conficker-B font facilement des scores de 50
  - Les conficker-C font facilement des scores de 200
  - Des comportements déviants apparaissent (non conficker)
    - Domaines étranges (r :iuet9uj.r2y51ov1 ou l48ks8pw :.27je)
    - Domaines infectieux (baldmanpower.com,kutlufamily.com, etc.)
- Sensibilité trop grande
  - Les serveurs sont à exclure
  - Le seuil est à remonter
  - Les blacklist seraient utiles.

Les domaines sans résolution

Le procédé étant trop sensible, on repart de la définition du conficker : génération de domaines aléatoires dont certains seront réservés "plus tard" quand le besoin se fera sentir.

- On exclut les serveurs (mais pour combien de temps, surtout pour les Microsoft ?)
- On remonte le seuil (à 100)
- On augmente la note de 5 points si le domaine ne résout pas.

Les domaines sans résolution : conclusion

L'amélioration est appréciable :

- Les faux positifs descendent drastiquement.
- Les conficker sont détectés plus vite

Mais :

- On est obligé d'attendre 20 requêtes pour repérer un conficker.
- On ne repère par les postes infectés par des pré-conficker.

Les blacklists

Suite aux travaux, entre autres, du conficker working group [4], la liste des domaines est disponible (avec ses collisions). On l'utilise.

- La blacklist conficker est intégrée (1,5 million pour avril)
- On ajoute une blacklist de malware (provenant d'une source référente ;-))
- On pondère chaque domaine :
  - 20 pour la liste du conficker (risque de post-réservation)
  - 5 pour la blacklist de malware (la liste est multiforme)
  - 50 si le domaine est avéré

[4] http://www.confickerworkinggroup.org

Les blacklists : conclusion

L'amélioration est latente :

- La distinction est facilitée
  - Conficker-C : note de 5000
  - Poste autre (Win32/Rbot.JMR par exemple) : détecté
  - Les faux positifs n'augmentent pas.

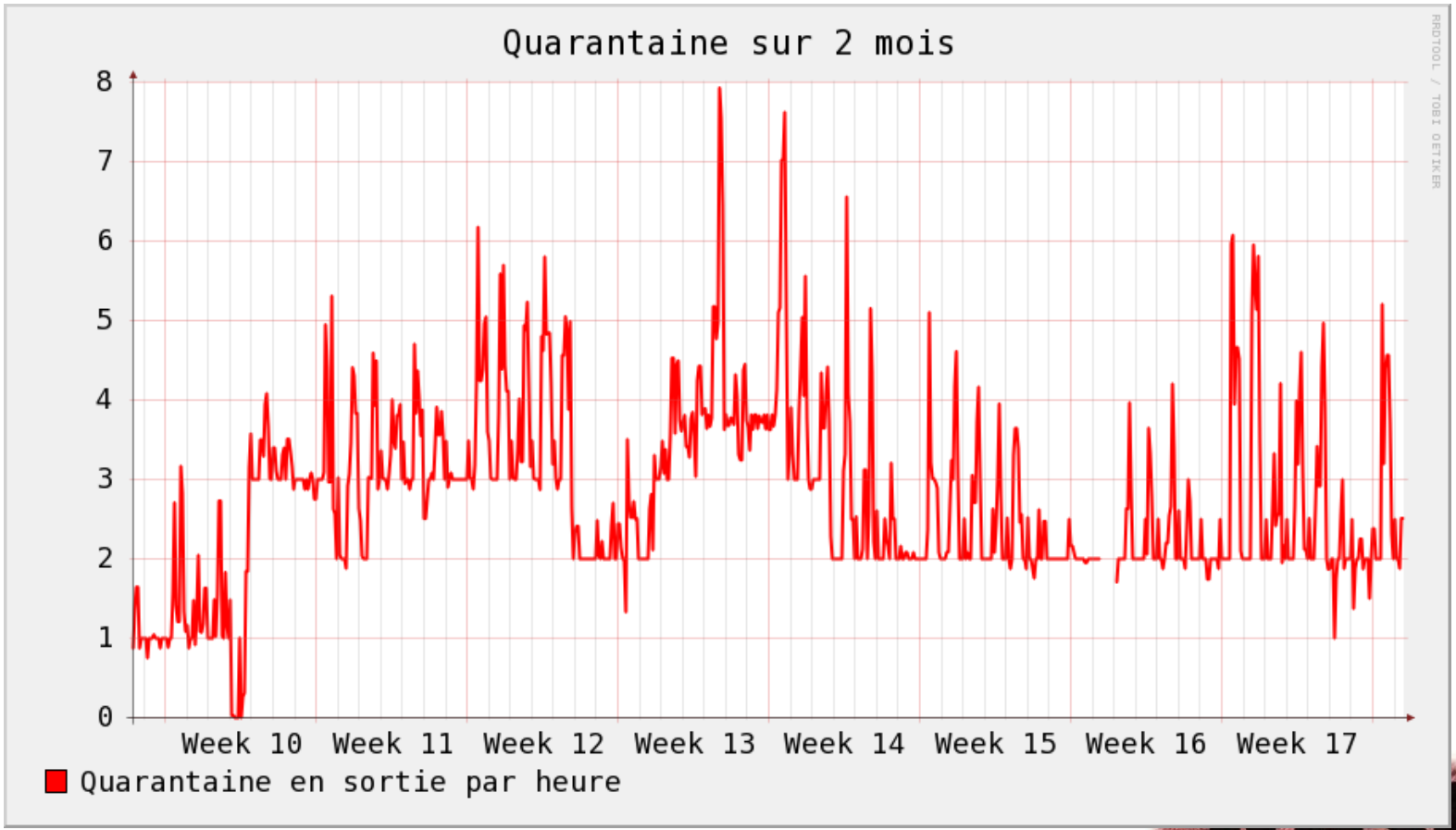
Et maintenant ?

Quarantaine

Processus habituel à l'UT1 :

- Mise en quarantaine
- Interception des communications web, avec message

Quarantaine : graphique



Mais ne pas oublier

- Les résolutions DNS doivent être locales (filtrage)
- Les pirates renforceront leurs méthodes de camouflage (plus de blacklist ?)
- La détection se fait sur des communications en UDP... (vive le spoofing !)

Des questions ?

Des questions ?